

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2003-208306
(P2003-208306A)

(43) 公開日 平成15年7月25日 (2003.7.25)

(51) Int.Cl. ⁷	識別記号	F I	テームコード*(参考)
G 0 6 F 9/30	3 1 0	G 0 6 F 9/30	3 1 0 C 5 B 0 1 3
	3 5 0		3 5 0 A 5 B 0 3 3
			3 5 0 F 5 B 0 8 1
9/38	3 7 0	9/38	3 7 0 X
9/45		9/44	3 2 2 A
審査請求 未請求 請求項の数11 O L (全 7 頁)			

(21) 出願番号 特願2002-347918(P2002-347918)
(22) 出願日 平成14年11月29日 (2002. 11. 29)
(31) 優先権主張番号 0 1 8 3 0 8 1 4 : 8
(32) 優先日 平成13年12月27日 (2001. 12. 27)
(33) 優先権主張国 欧州特許庁 (E P)

(71) 出願人 591002692
エスティーマイクロエレクトロニクスエ
ス. アール. エル.
STMicroelectronics
S. r. l.
イタリア国 ミラノ 20041 アグラータ
ブリアンツァ ヴィア ツィー オリヴ
エッティ 2
(74) 代理人 100072051
弁理士 杉村 興作 (外1名)

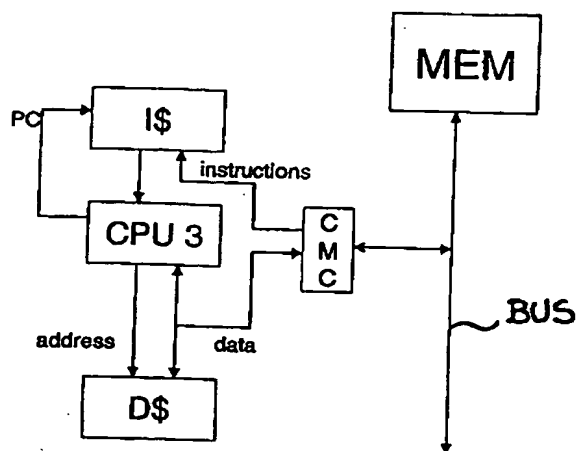
最終頁に続く

(54) 【発明の名称】 プロセッシングアーキテクチャ、該アーキテクチャ実現システム及び該システムのオペレーション方法

(57) 【要約】 (修正有)

【課題】 2以上の異なる命令セットを実行し得るプロセッシングアーキテクチャの提供。

【解決手段】 このアーキテクチャは、第1セットの命令 (OsTask1.1, OsTask1.2, . . .) と第2セットの命令 (MmTask2.1, MmTask2.2, MmTask2.3, . . .) の両方を実行するよう構成された単一CPUを具える。該単一CPUは、該CPUが前記第1セットの命令 (OsTask1.1, OsTask1.2, . . .) を実行する第1動作モードと該CPUが前記第2セットの命令 (MmTask2.1, MmTask2.2, MmTask2.3, . . .) を実行する第2動作モードとの間で切り換えるように構成されている。この解決方法は種々のCPUに対する2つ以上の実行モード間で複数の切り換え命令を用いることにより一般化することができる。



【特許請求の範囲】

【請求項1】 第1のCPU（CPU1）により実行されるようにコンパイルされ第2のCPU（CPU2）では実行できない少なくとも一つの第1の命令セット（OsTask1.1, OsTask1.2, . . .）と第2のCPU（CPU2）により実行されるようにコンパイルされ第1のCPU（CPU1）では実行できない少なくとも一つの第2の命令セット（MmTask2.1, MmTask2.2, MmTask2.3, . . .）を実行するプロセッシングアーキテクチャであ

って、該アーキテクチャは、前記第1セットの命令（OsTask1.1, OsTask1.2, . . .）と前記第2セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, . . .）の両方を実行するよう構成された単一プロセッサ（CPU3）を具え、該単一プロセッサ（CPU3）は、該プロセッサが少なくとも前記第1セットの命令（OsTask1.1, OsTask1.2, . . .）を実行する第1動作モードと該プロセッサが前記第2セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, . . .）を実行する第2動作モードとの間で選択的に切り換え可能であり、前記単一プロセッサ（CPU3）は、少なくとも前記第1動作モードと前記第2動作モードとの間で少なくとも一つの切り換え命令（Model_NotMode2 flag）を認識し、前記少なくとも一つの切り換え命令に従って前記第1動作モードと前記第2動作モードとの間の切り換えを実行するよう構成することを特徴とするプロセッシングアーキテクチャ。

【請求項2】 前記単一プロセッサ（CPU3）はデータ用の単一キャッシュ（DS）に関連することを特徴とする請求項1記載のアーキテクチャ。

【請求項3】 前記単一プロセッサ（CPU3）は命令用の単一キャッシュに関連することを特徴とする請求項1又は2記載のアーキテクチャ。

【請求項4】 前記単一プロセッサ（CPU3）はバスを介してメインメモリと対話する単一インタフェース（CMC）に関連することを特徴とする請求項1-3の何れかに記載のアーキテクチャ。

【請求項5】 メモリ内の前記命令をアドレスするために単一プログラムカウンタが設けられていることを特徴とする請求項1-4の何れかに記載のアーキテクチャ。

【請求項6】 前記単一プログラム（CPU3）は、前記第1セットの命令（OsTask1.1, OsTask1.2, . . .）及び前記第2セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, . . .）をそれぞれデコーディングする少なくとも一つの第1のデコーディングモジュール（DEC1）及び少なくとも一つの第2のデコーディングモジュール（DEC2）を具えることを特徴とする請求項1-5の何れかに記載のアーキテクチャ。

【請求項7】 前記第1セットの命令（OsTask1.1, OsTask1.2, . . .）及び前記第2セットの命令（MmTask2.1, MmTask2.2, MmTask2.3, . . .）のオペランドを読み出すためのレジスタの統合ファイル（レジスタファイ

ル）を具えることを特徴とする請求項1-6の何れかに記載のアーキテクチャ。

【請求項8】 前記第1の動作モード又は前記第2の動作モードにおいて命令の実行に使用されないとき選択的に非活性化し得るユニットを具えることを特徴とする請求項1-7の何れかに記載のアーキテクチャ。

【請求項9】 請求項1-8の何れかに記載のアーキテクチャを実現するプロセッシングシステム。

10 【請求項10】 請求項9に記載のプロセッシングシステムを使用する方法であって、

前記少なくとも一つの第1の命令セット（OsTask1.1, OsTask1.2, . . .）の命令のセットと前記少なくとも一つの第2の命令セット（MmTask2.1, MmTask2.2, MmTask2.3, . . .）の命令のセットをコンパイルするオペレーションと、

前記少なくとも一つの切り換え命令（Model_NotMode2 flag）を前記命令のセットの先頭に付与するオペレーションと、を具えることを特徴とするプロセッシングシステムの使用方法。

20 【請求項11】 前記第1のCPU（CPU1）及び前記第2のCPU（CPU2）のコンパイルフローをそのまま用いて各プロセスをコンパイルし、前記切り換え命令（Model_NotMode2 flag）を前記命令のセットの先頭に入力することを特徴とする請求項10記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プロセッシングアーキテクチャ及び該アーキテクチャを実現するシステムに関するものである。本発明は特に移動通信システムに使用し得るマイクロプロセッシングアーキテクチャに注目して開発したものである。しかし、本発明の範囲はこの応用分野に限定されるものではない。

【0002】

【従来技術】セルフォンの代表的なシステムアーキテクチャは複数の中央処理装置（CPU）のアーベイラビリティに基づいている。これらのCPUは通常2つであり、各々特定の目的を満たす。第1のCPUはオペレーティングシステムの制御機能に実質的に類似する制御機能を実行する。このタイプの用途は計算上の観点からの厳しい要求も、高い性能も要求しない。通常、簡単なフェッチ-デコード-リード-実行-ライトバックのステージからなるスケーラパイプライン形のアーキテクチャの使用を想定している。第2のCPUは計算上の約束及び性能に関して相違する特性を有する機能を実行する。この理由のために、通常、スーパースケーラプロセッサ又は1サイクルにつき複数の命令を発行し実行することができるベリロンギンストラクションワード（VLIW）パイプラインプロセッサの使用を想定している。これらの命令は（VLIWアーキテクチャの場合には）コンパイルステージで、（スーパースケーラプロセッサの場合に

3

は) 実行ステージでスケジューリングすることができる。

【0003】この計算リソースの重複はメモリに関する要件の重複をまねき、その結果として大きな電力消費を生ずる。大きな電力消費は2つのCPUの一方又は他方を交互にスリープモードに設定することにより部分的に制限できるが、避けることはできない。

【0004】図1につき説明すると、上述したタイプのワイヤレス応用に対する代表的なアーキテクチャはCPU1及びCPU2で示す2つのマイクロプロセッサのよう

な2つのCPUを具え、各自キャッシュメモリアーキテクチャを具えている。

【0005】CPU1は代表的には32ビットパイプラインスーパーマイクロプロセッサである。これは、その内部アーキテクチャが種々のロジックステージからなり、各ロジックステージが命令を一つの特定の状態で含むことを意味する。この状態は下記の状態：

- メモリからの命令のローディング；
- デコーディング；
- レジスタファイルのアドレッシング；
- 実行；及び

-メモリからのデータの書込み/読み出し；

の一つである。ビット数はCPU1が動作するデータ及び命令の長さに関連する。命令はコンパイルにより特定の順序で発生され、その順序で実行される。

【0006】CPU2は代表的には128ビットパイプラインスーパーマイクロプロセッサ又はVLWマイクロプロセッサである。これは、その内部アーキテクチャが種々のロジックステージからなり、いくつかのロジックステージが、例えば実行ステップで、命令を並列に実行することができることを意味する。代表的には、4つの32ビット命令(128ビットに相当する)を並列に実行し、データは32ビットで表現する。並列に実行可能な複数の実行ステージを供給するために命令を実行中に動的に順序付ける場合及び命令が相互に依存せず、従ってソースコードのコンパイルにより静的に発生される順序を変更する場合には、そのプロセッサはスーパーバスケーラであるという。

【0007】その代わりに、命令をコンパイルステップにおいて静的に順序付けし、固定の順序で実行する(実行中に変更不能)場合には、そのプロセッサはVLW(ベリーロングインストラクションワード)という解決手法に対応する。

【0008】再び図1に付き説明すると、各プロセッサCPU1、CPU2は各自DSで示すデータキャッシュとISで示す命令キャッシュを有するため、メインメモリMEMから処理すべきデータと実行すべき命令を並列にロードすることができる。2つのプロセッサCPU1、CPU2はシステムバスにより相互接続されるとともに、このシステムバスによりメインメモリMEMに接

(3)

特開2003-208306

4

続される。2つのプロセッサCPU1、CPU2はバスへのアクセスを競争する。このアクセスは、これらのプロセッサが実行すべき命令又はデータ又はその両方がメインメモリ内に位置し、各自のキャッシュにないときに、コアメモリコントローラCMCというそれぞれのインタフェースを介して達成される。このようなシステムは2つのマイクロプロセッサを使用し、それぞれに対応する2つのメモリ階層が不可欠であり、占有面積及び電力消費の点から幾分費用がかかることが理解される。一例として、代表的な応用では、CPU1は通常16キロバイトのデータキャッシュと16キロバイトの命令キャッシュを有し、CPU2は32キロバイトのデータキャッシュと32キロバイトの命令キャッシュを有する。

【0009】図2はCPU1のロジックスキームを示す。第1ステップは実行すべき命令が関連する命令キャッシュISのメモリアドレスを発生する。このアドレス(プログラムカウンタという)は命令のローディング(Fetch)を生じさせ、フェッチした命令をデコードし(Decode)、オペランドをアドレスするビットフィールドから機能(例えば、レジスタファイルに位置する2つのレジスタ内の2つの値の加算)を定義するビットフィールドを分離する。オペランドアドレスはレジスタファイルに送られ、これから命令のオペランドを読み出す。オペランドと実行すべき命令を定義するビットは実行ユニット(Execute)に送られ、実行ユニットが所望の演算(例えば加算)を実行する。次いでその結果はレジスタファイル内のメモリにリストアされる(Writeback)。

【0010】その代わりに、ロード/ストアユニットが、メモリデータの読出し/書込みをこの目的専用の特定の命令を用いて行うことができる。他方、命令セットと(マイクロ)プロセッシングアーキテクチャとの間に一対一の一義的対応(biunivocal)が存在することが容易に認識される。

【0011】CPU1について述べたことはCPU2にもほぼあてはまり、図3に同様の表現で示されている。主要な違いは、CPU2の場合には、スーパーバスケーラ及びVLWプロセッサで並列に動作し得る使用可能な実行ユニットが多数ある点にある。この点に関し、図3にはExcute2.1, Excute2.2, . . . , Excute2.nにより示された種々のステージが示されている。しかし、この場合にも、命令セットとプロセッシングアーキテクチャとの間に一対一の一義的対応が存在する。例えばワイヤレスプロセッサのアーキテクチャのようなアーキテクチャでは、2つの命令セットが相違することが一般に見られる。このことは、CPU1で実行される命令をCPU2で実行できず、その逆も同様であることを意味する。

【0012】図4及び図5に関して、このような2つの各別の命令セットの形を取る複数のタイプのプロセシスを処理する必要があるものとする。例えば、先に述べた

5

応用コンテキスト（移動通信）に関して、2つのタイプのプロセス：

—CPU1で実行されるオペレーティングシステムプロセスに類似するプロセスOsTask1.2、OsTask1.2、等；及び

—CPU2で実行されるコンテンツ（通常オーディオ/ビデオ/グラフィックコンテンツのようなマルチメディアコンテンツ）のプロセッシングに関するプロセスMmTask2.1、MmTask2.2、MmTask2.3、等；に区別することができる。

【0013】前者のプロセスはCPU1のコンパイラにより発生される命令を含み、従ってCPU1自身で実行することができるがMCPU2で実行することはできない。第2のプロセスに対しては、正確にその逆が当てはまる。更に、各CPUは各自のコンパイルフローで特徴付けられ、各CPUのコンパイルフローは他の使用CPUのコンパイルフローと独立である点に注意されたい。

【0014】図5は、上記のタスクのスケジューリングのシーケンスを2つのプロセッサCPU1及びCPU2にどのように分配するかを示す。上記のプロセスの総実行時間を100とした場合、前者が総実行時間の10%を占め、後者が90%を占めるのが代表的である。これから、CPU1は時間の10%しか動作しないので、CPU1は時間の90%が冗長となることになる。

【0015】上述の特性を利用してCPU1をターンオフすることによりエネルギーの節約を達成することができる。しかし、パワーダウンプロシージャはプロセッシングの追加の待ち時間を導入し、この時間が上述の10%に付加される。これらのプロシージャは実際には、レジスタファイルの全ての内部レジスタ並びにコア内に存在する他のユニット（例えばデコーディングユニット、実行ユニット）に供給するクロックをゲートすることによりレジスタファイルを除くCPUのパワーダウン；

—CPUの完全なパワーダウン、キャッシュメモリへのエネルギー供給は維持；

—CPU全体並びにデータキャッシュ及び命令キャッシュのパワーダウン；を含む。

【0016】構造上の点から見ると、上述したオペレーションに続いてプロセッサのパワーを再投入する時にパワーダウン前のプロセッサ自体を特徴づけるプロセッサの状態を復元する必要があるため、導入される待ち時間は数十ナノ秒から数十/数百ミリ秒の範囲になる。従って、上述のパワーダウンプロシージャはエネルギーの点からも計算の点からも効果的でない。

【0017】

【発明が解決しようとする課題】従って、本発明の目的は、上述した欠点を克服することができるマイクロプロセッシングシステムアーキテクチャを構成することにあ

(4)

特開2003-208306

6

る。

【0018】

【課題を解決するための手段】本発明によれば、この目的は特許請求の範囲において特定した特徴を有するアーキテクチャによって達成することができる。本発明は対応するシステム、並びに対応する使用方法にも関する。

【0019】本発明の解決方法は、本質的には、上述した態様に従うオペレーティングに予想される制御コードをサポートするのに必要とされるリソース（CPU、メモリ等）の二重化、一般に多重化は、最初から想定されている2つ（又はそれ以上）のCPUを単一の最適化された（マイクロ）アーキテクチャ、即ち種々のCPUのコンパイラにより発生された命令を実行し得る単一の新しいプロセッサに融合させることができれば避けることができるという認識に基づくものである。この場合には、前記新プロセッサは1以上の特別命令をデコードし、例えばその機能を異なる命令セットに固有の2以上の実行モードの間で切り換えることができなければならないという唯一の要件がある。

【0020】この命令又はこれらの命令はCPUに予め関連するコンパイラを用いてコンパイルされた命令の各セットの先頭に入れる。特に2つのステップが想定される。第1のステップはCPU1又はCPU2のコンパイルフローを（そのまま）用いて各プロセスをコンパイルする（以後、本発明は任意の数のこのようなCPUに適用し得るが、説明を簡単にするために2つの出発CPUについて説明する）。第2のステップは各命令セットを取り、その先頭に特別命令を入力し、最適化されたマイクロアーキテクチャのフレームワークにおけるCPU1の実行モードとCPU2の実行モードとの間のモード切り換えを信号し許可する。

【0021】以上の解決方法はメモリ及び電力消費に関し著しい節約をもたらす。更に、切り換え命令を検出する1つのフェッチユニット、（2つのCPU、CPU1及びCPU2の各々に対する）2つのデコーディングユニット、単一のレジスタファイル、複数の実行ユニット、及びロード/ストアユニット（特別命令が検出されると、構成される）を使用するのみとすることができる。

【0022】

【発明の実施の形態】本発明を図面を参照して実施例につき詳細に説明するが、本発明はこのような実施例に限定されるものではない。

【0023】本発明の模範的な実施例の詳細な説明 上述したように、本発明が基づく主な思想は、低い計算ウェイト（例えば時間の10%）のプロセスの実行をサポートするために、プロセッシングリソースの二重化は不要であるという認識にある。図6に図式的に示すように、本発明の解決方法は、既知の解決方法ではCPU1及びCPU2のような2以上の個別のCPUで実行するよう

50

7

設計されたプロセスをアプリケーションによらずに実行し得るCPU3で示す新しいプロセッサ又はCPUアーキテクチャを定義するものであり、新しいアーキテクチャに対しリコンパイルする必要がある。

【0024】基本的には、本発明の解決方法のねらいは、各CPUに想定されたもとのコンパイルフローを再利用し、その下流に、対応するプロセスの実行をコンパチブルにする第2ステップを付加することにある。

【0025】特に、図7を参照して、第1のコンパイルステップにおいて、オペレーティングシステム用のプロセスOsTask1.1のソースコードを考察する。図1に示すような伝統的なアーキテクチャでは、対応する命令は対応するコンパイラを用いてCPU1で実行する必要がある。次に、同じ第1ステップにおいて、マルチメディアオーディオ/ビデオ/グラフィックアプリケーション用のプロセス(MmTask2.1)のソースコードのコンパイルを考察する。このプロセスは、図1に示すような伝統的なアーキテクチャでは、この場合にもCPU1のコンパイラと異なる対応するコンパイラを用いてCPU2で実行される。更に、図1に示すようなスキームでは、2つのプロセッサCPU1及びCPU2が独立の命令セットのアーキテクチャを有する点を思い出す必要がある。

【0026】次に、第2のステップを考察する。このステップでは（少なくとも）一つの新しい特別命令を丁度発生された命令の頭に入力する。この特別命令は対応する命令セットに続く命令の所属を識別することができる。従って、この特別命令はCPU3がCPU1の命令セットの実行モードからCPU2の命令セットの実行モードに、及びその逆に切り換えることができるようにする手段を表す。

【0027】図8は、図1のアーキテクチャを、CPU3で示す単一のCPUに、関連するキャッシュメモリ、即ちデータキャッシュメモリDS及び命令キャッシュメモリISを設けることによりどのように巨視的観点から簡略化できるかを示す。このように、対応するメモリサブシステムはキャッシュメモリの二重化を必要とせず、対応するバスにインタフェースするインタフェースCMCを経るメインメモリMEMへのアクセス要求の競合から開放される。これから性能の明らかな改善が得られる。

【0028】他方、プロセッサCPUは、対応するコンパイラにより発生されCPU1のタイプのプロセッサで実行すべき命令もCPU2のタイプのプロセッサで実行すべき命令も実行できなければならない。これは同様に、2つのCPU間の実行モードの制御命令も実行できることを想定している。

【0029】図9はここに提案するCPU3のロジックスキームを示す。命令はメモリにて単一プログラムカウンタによりアドレスされ、Fetch & Alignで示すユニットによりロードされる。このユニットは次に命令をCP

(5)

特開2003-208306

8

U1及びCPU2の命令セットにコンパチブルなデコーディングユニットに送る。これらのデコーディングユニットは両方とも、命令セット1に対する実行モードを命令セット2に対する実行モードに及びその逆に切り換える特別命令の存在を検出することができる。こうして活性化されたフラグをCPU内に存在する全てのユニットに送って、そのCPU1-コンパチブルモード又はCPU2-コンパチブルモードのオペレーションを構成する。特に、図9において、このフラグはModel_NotMode2 flagとして示す信号と同一である。最も簡単な実施例では、このフラグは、CPU3がCPU1の命令セットで動作するとき論理値“1”を有し、CPU3がCPU2の命令セットで動作するとき論理値“0”を有する。これと反対にすることもできること勿論である。

【0030】ロードされた後続の命令を（Dec1及びDec2で示すステージで）デコードし、機能（例えば加算）を定義するビットフィールドをオペランドをアドレスするビットフィールドから分離する。対応するアドレスをレジスタファイルに送り、レジスタファイルから命令のオペランドを読み出す。これらのオペランドと実行すべき機能を定義するビットを要求されたオペレーションを実行する多数の実行ユニット（Execute1. . . . , Execute2.2. , Execute+1. . . . , Execute. . . .）に送る。次に、結果を図2及び図3と同様にライトバックステージでレジスタファイルにリストアすることができる。その代わりに、ロード/ストアユニットがメモリから/メモリへのデータの読出し/書込みを行うことができ、各オペレーティングモードにはこの目的用の命令が存在する。

【0031】特に、現在使用されていない実行モードとコンパチブルであるユニット（例えば、デコーディングユニットDec1及びDec2）は電力を消費しないように適切に“ターンオフ”することができることが認識されるであろう。

【0032】本発明の原理を損なうことなく、上述した構成及び実施例の細部は特許請求の範囲に限定された本発明の範囲から逸脱することなく広範囲に変更することができること勿論であり、特に本発明の解決方法は種々のCPUに対する2以上の実行モードの間で複数の切り換え命令を用いることにより一般化することができること明らかである。

【図面の簡単な説明】

【図1】 2以上のCPUを用いる従来のプロセッシングシステムの構成図である。

【図2】 図1の従来のプロセッシングシステムのCPU1のロジックスキームを示す図である。

【図3】 図1の従来のプロセッシングシステムのCPU2のロジックスキームを示す図である。

【図4】 CPU1で実行されるタスクとCPU2で実行されるタスクの実行時間の比較を示す図である。

10

20

30

40

50

(6)

特開2003-208306

9

【図5】 図4のタスクが2つのプロセッサCPU1及びCPU2にどのように分配されるかを示す図である。

【図6】 本発明によるCPUアーキテクチャを示す概念図である。

【図7】 本発明によるアーキテクチャにおけるコンパイルステップを示す図である。

【図8】 本発明によるアーキテクチャをブロック図で示したものである。

10

【図9】 図8に示すアーキテクチャの構造の詳細とオペレーションの詳細を示す図である。

【符号の説明】

CPU3 単一CPU

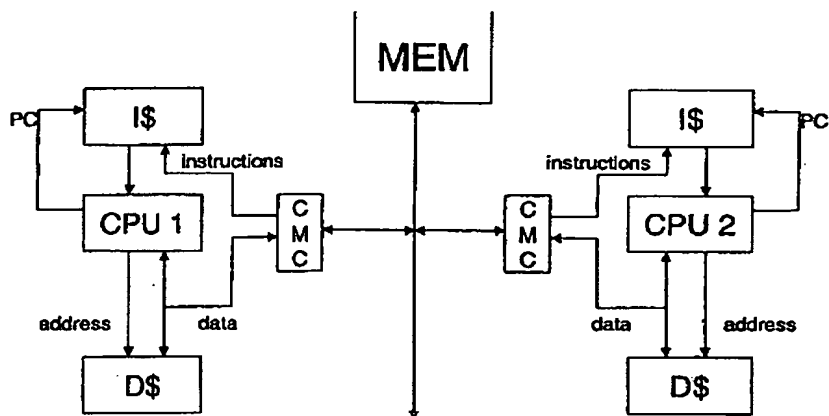
IS 命令キャッシュメモリ

DS データキャッシュメモリ

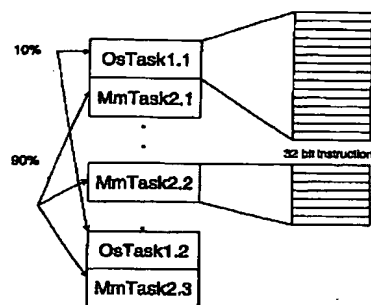
MEM メインメモリ

CMC インタフェース

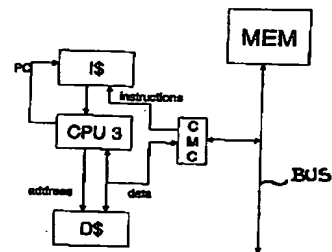
【図1】



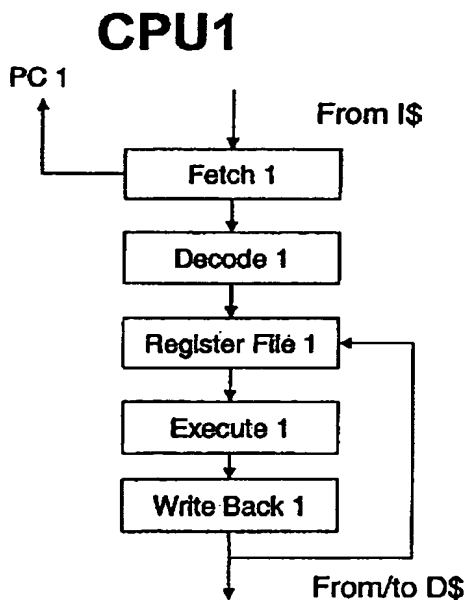
【図4】



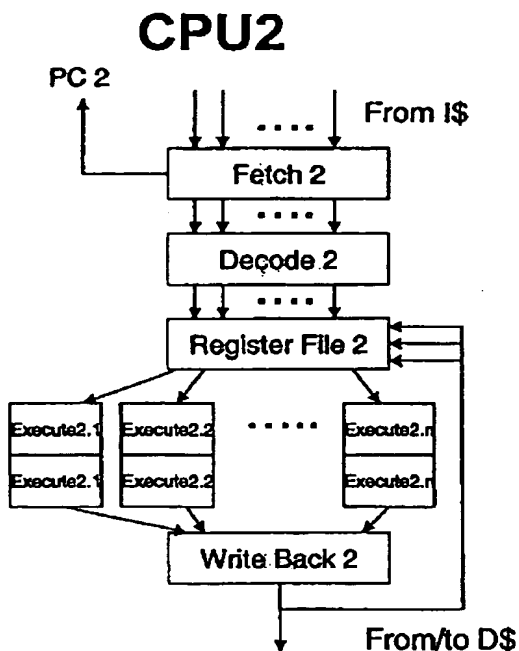
【図8】



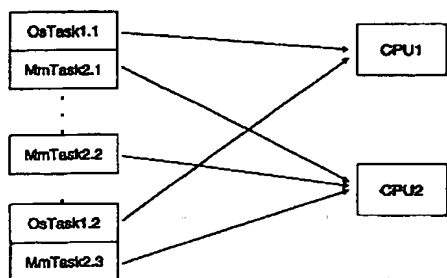
【図2】



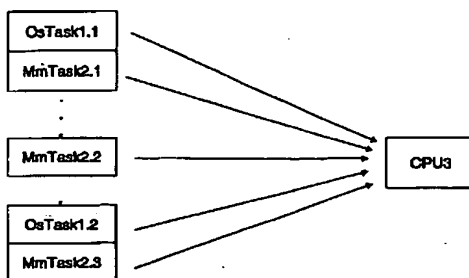
【図3】



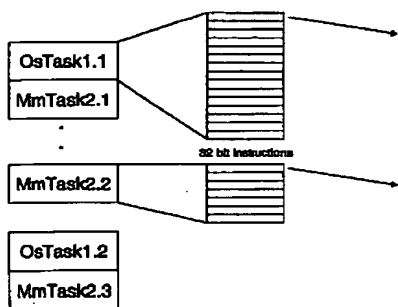
【図5】



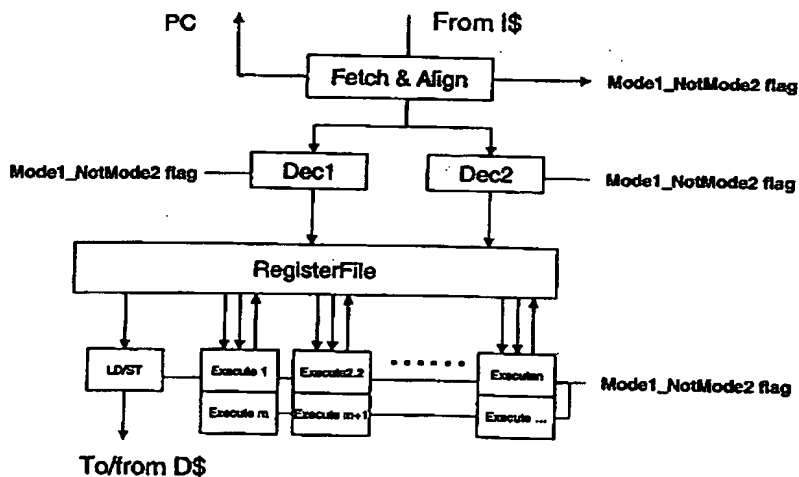
【図6】



【図7】



【図9】



フロントページの続き

(72) 発明者 アレッサンドロ クレモネージ
 イタリア国 ロディ 20079 エッセ ア
 ンジェロ ロディジャーノ ヴィア ダイ
 ツ 102
 (72) 発明者 ファブリツィオ ロヴァティ
 イタリア国 ミラノ 20092 チニゼッロ
 バルサモ ヴィア コッレオーニ 15

(72) 発明者 ダニーロ パウ
 イタリア国 ミラノ 20099 セスト サ
 ン ジョヴァンニ ヴィア ダンテ 131
 Fターム(参考) 5B013 DD04
 5B033 AA05 AA10 AA14 AA15 BA05
 BE00 BE05
 5B081 CC00 CC53